RSAC 2025 Conference

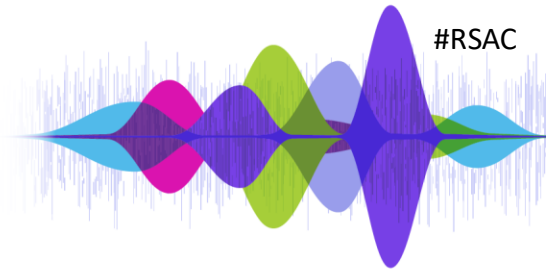
Many Voices.
One Community.

SESSION ID: BR-W09

That's Not My Signature! Fail-Stop Signatures for a Post-Quantum World

Hila Dahari-Garbian

PhD student
Weizmann Institute of Science



Disclaimer

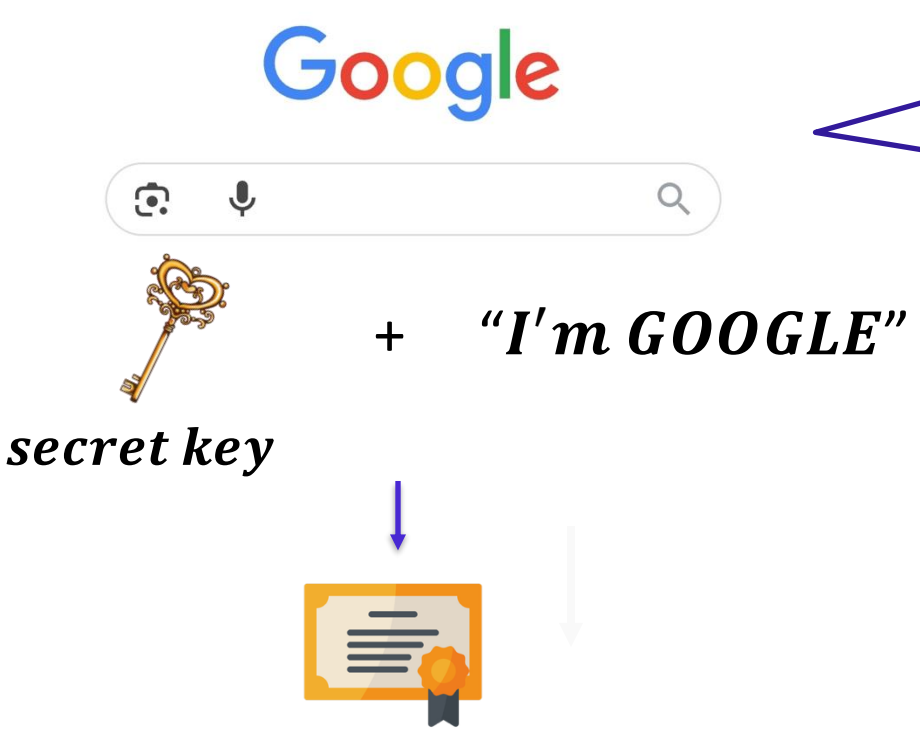
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Digital Signatures

- Digital signature allows verification of:
  - The **content** of a message
  - The **identity** of the signer



RSAC 2025 Conference

Weaponizing Signatures

- In 2011, forged certificates signed by **DigiNotar** exploited for MiTM attacks against Gmail users in Iran

Weaponizing Signatures

- In 2012, forged signatures signed by **Microsoft** exploited by the **Flame** malware for a large scale cyber-espionage and sabotage campaign

Weaponizing Signatures

- **Two signature forgery attacks, but are they the same?**

DigiNotar®
A VASCO COMPANY

https://

Google

Compromising **secret key**

DigiNotar attack [2011]:

Cyber attack: Hackers gained **internal access**

Compromising **secret key**

Breaking the cryptographic **primitive**

DigiNotar attack [2011]:

Cyber attack: Hackers gained **internal access**

Flame attack [2012]:

Cryptanalytic attack: Hackers broke the **MD5 cryptographic primitive**

Compromising **secret key**

Breaking the cryptographic **primitive**

DigiNotar attack [2011]:

Cyber attack: Hackers gained **internal access**

DigiNotar cannot be trusted!!!

Compromising **secret key**

Breaking the cryptographic **primitive**

MD5 is broken!!!

Flame attack [2012]: Cryptanalytic attack: Hackers broke the **MD5 cryptographic primitive**

Compromising **secret key**

Breaking the cryptographic **primitive**

It is crucial to distinguish between the attacks!

However, this is not currently possible with modern digital signature schemes

Nigel Smart
26 mins ·

Quantum update. LWE is definitely still alive. Or at least still alive until someone opens the box.

CRYPTOMATHiC

WHY CRYPTOMATHIC? WHAT WE OFFER ABOUT INSIGHTS CONTACT

4 MIN READ

NIST PQC FINALISTS UPDATE: IT'S OVER FOR THE RAINBOW

Edlyn Teske : 26. March 2022

Quantum Computing Report
Where Qubits Entangle with Commerce

NIST PQC Round 4 Candidate SIKE May Have Been Broken

We reported last month that NIST had selected four Post-
Round 3 and another four algorithms were selecte

Breaking Rainbow Takes a Weekend on a Laptop

Ward Beullens

IBM Research, Zurich, Switzerland
wbe@zurich.ibm.com

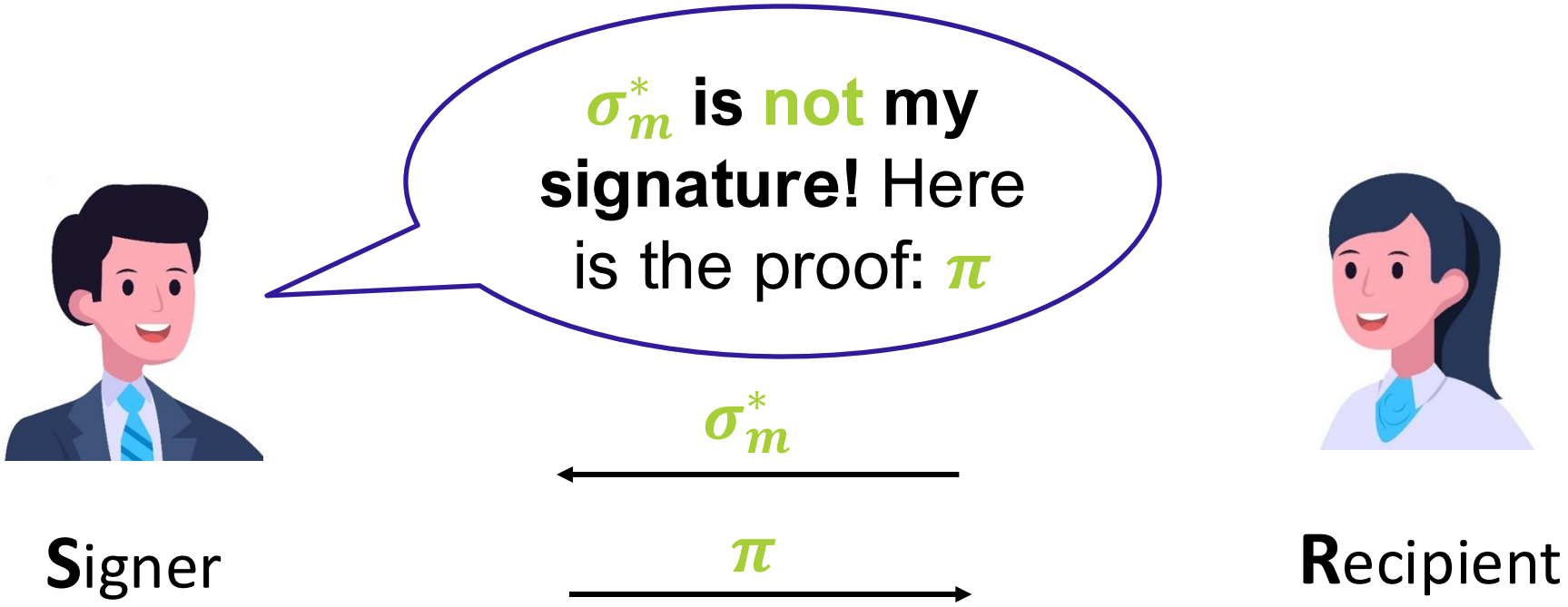
against the
t signature
dardization
s for all the

The next breakthrough may not be published…

If a **digital signature** is **broken**,
and a forgery was created,
can we **identify** this?

Fail-Stop Signature (FSS)

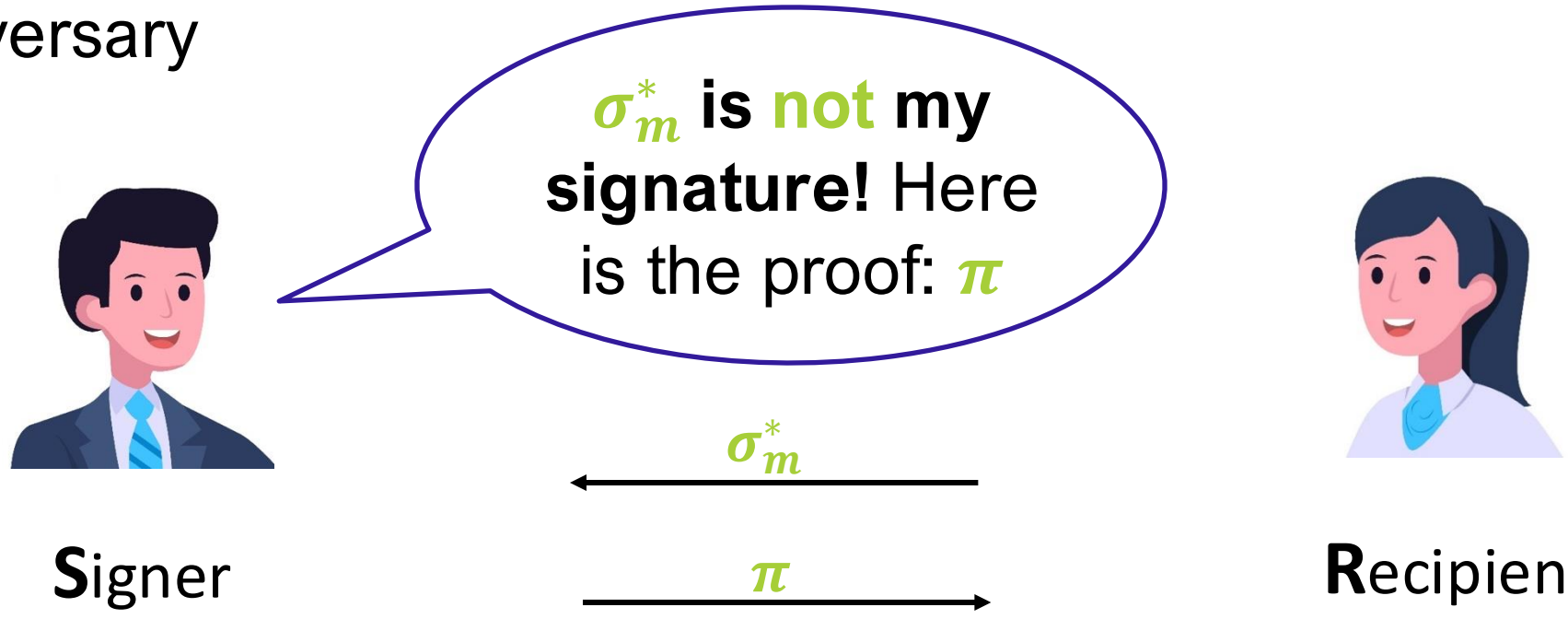
- **Fail-Stop Signature** [PW89] has a "**forgery detection**" procedure



RSAC 2025 Conference

Fail-Stop Signature (FSS)

- We can prove a forgery has occurred
- We cannot take back a signature
- Unbounded adversary

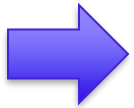


Related Works

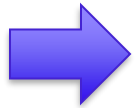
- Lower bounds on FSS [**H**eijst**P**edersen**P**fitzmann**93**]
- Constructions based on standard assumptions (Factoring, RSA, OWF, etc.): [**H**eyst**P**edersen92, **D**amgard**P**edersen**P**fitzmann**94**, **B**ari**P**fitzmann**97**, **S**afavi-**N**aini**S**usilo**W**ang**20**]
- Proof-of-forgery for hash-based signature [**K**iktenko**K**udinov**B**ulychev**F**edorov**21**]

Our Work: Fail-Stop for a Post-Quantum World

Fail-Stop Signature [PW89] **Unbounded adversaries**

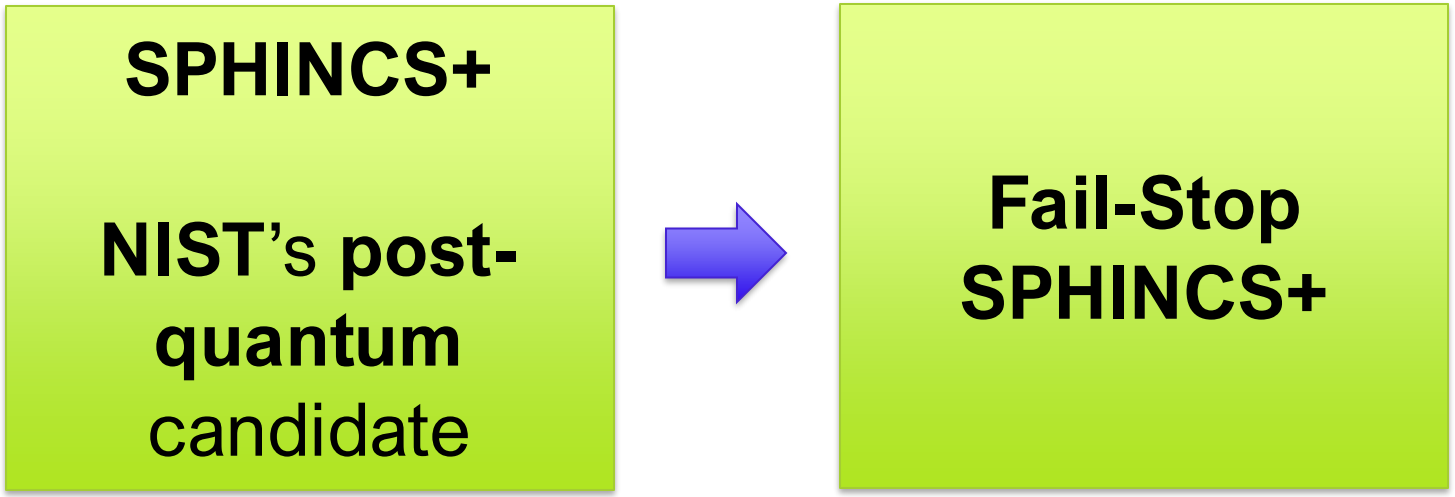


Introduce the "**fine-grained** adversarial model"



Fail-Stop Signature **Powerful but bounded adversaries**

Our Construction



- SPHINCS+ has Lamport signature as basic structure

Lamport One-Time Signature

- The **security** **assumption**: one way function $\boldsymbol{h}$: $\{0,1\}^{\mathbf{128}} \rightarrow \{0,1\}^{\mathbf{128}}$
- Message: 0 or 1

Signer

(r_0, r_1)

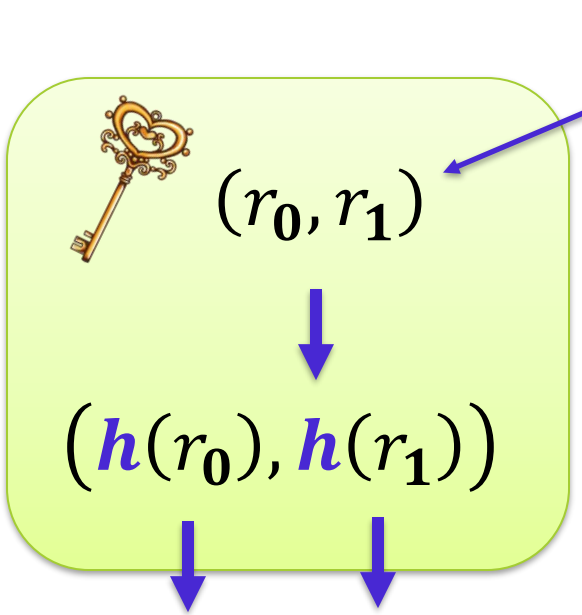
$\boldsymbol{sk}$

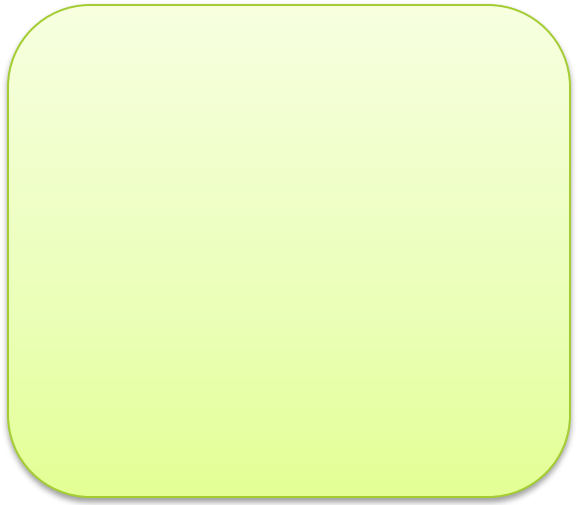
Recipient

Lamport One-Time Signature

- The **security** **assumption**: one way function $\boldsymbol{h}: \{0,1\}^{\mathbf{128}} \rightarrow \{0,1\}^{\mathbf{128}}$
- Message: 0 or 1

Signer





Recipient

Lamport One-Time Signature

- The **security** **assumption**: one way function $\boldsymbol{h}: \{0,1\}^{\mathbf{128}} \rightarrow \{0,1\}^{\mathbf{128}}$
- Message: 0 or 1

Signer

(r_0, r_1)

$(h(r_0), h(r_1))$

(vk_0, vk_1)

message:

$\mathbf{0/1}$ **signature**

$\sigma_b = (b, r_b)$

Recipient

Lamport One-Time Signature

- The **security** **assumption**: one way function $\boldsymbol{h}: \{0,1\}^{\mathbf{128}} \rightarrow \{0,1\}^{\mathbf{128}}$
- Message: 0 or 1

Signer

(r_0, r_1)

$\big(h(r_0), h(r_1)\big)$

(vk_0, vk_1)

message:

0/1 **signature**

$\sigma_b = (b, r_b)$

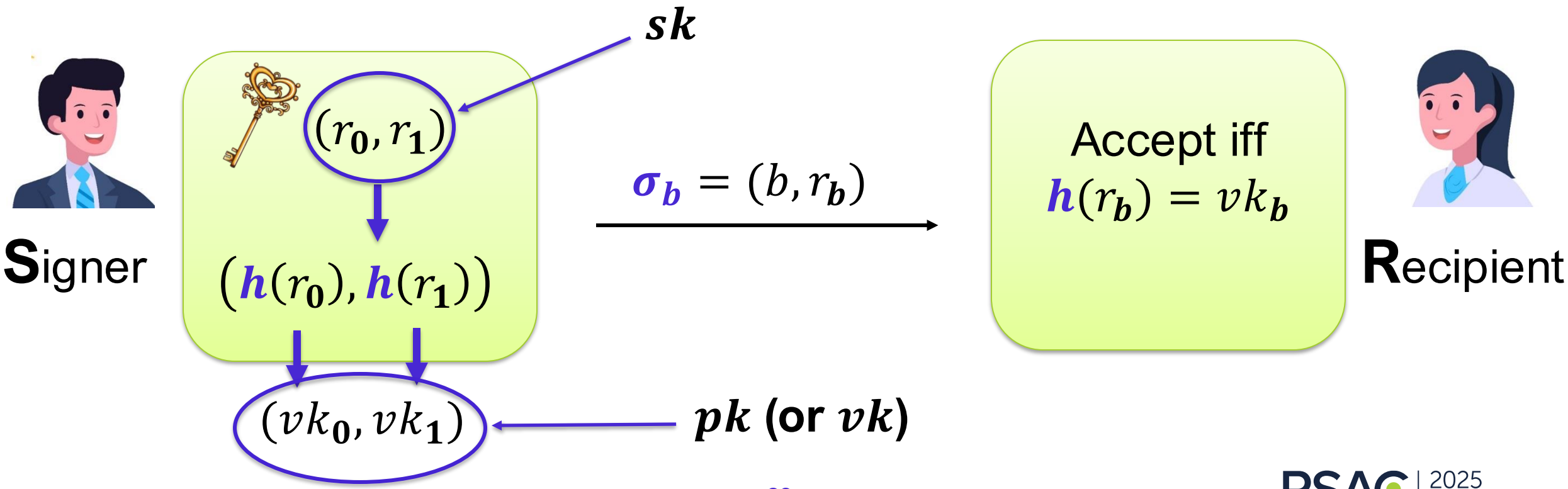
Accept iff

$h(r_b) = vk_b$

Recipient

Lamport One-Time Signature

- **Necessary condition for FSS:** each vk corresponds to **multiple** sk



RSAC 2025 Conference

Lamport One-Time Signature

- **[KKBF21]**: use a compressing $h: \{0,1\}^{128+c} \to \{0,1\}^{128}$

$$expect\ \sim 2^c\ r^*\ such\ that\ h(r^*) = h(r)\ for\ every\ r$$

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b = (b, r_b)$ $\longrightarrow$

Accept iff
$h(r_b) = vk_b$

Recipient

Lamport One-Time Signature

- What is the **proof-of-forgery?**

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b = (b, r_b)$ $\longrightarrow$

Accept iff
$h(r_b) = vk_b$

Recipient

Lamport Fail-Stop Signature

- **Collision-resistance (CRH)** hash function:

$$\boldsymbol{h}: \{0,1\}^{\mathbf{128}+c} \rightarrow \{0,1\}^{\mathbf{128}}$$

- It is hard for a **PPT** adversary $\boldsymbol{A}$ to find two different x_0, x_1 such that

$$x_0 \neq x_1,$$

$$\boldsymbol{h}(x_0) = \boldsymbol{h}(x_1)$$

Lamport Fail-Stop Signature

- Use a **compressing-collision-resistance** $h: \{0,1\}^{128+c} \rightarrow \{0,1\}^{128}$

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b = (b, r_b)$ $\longrightarrow$

Accept iff
$h(r_b) = vk_b$

Recipient

Lamport Fail-Stop Signature

$\sigma_b^* = (b, r_b^*)$

Adversary

Signer

(r_0, r_1)

$(h(r_0), h(r_1))$

(vk_0, vk_1)

Recipient

Lamport Fail-Stop Signature

Adversary

$\sigma_b^* = (b, r_b^*)$

Signer

(r_0, r_1)

$(h(r_0), h(r_1))$

(vk_0, vk_1)

σ_b^*

$h(r_b^*) = vk_b$

Accept

Recipient

Lamport Fail-Stop Signature

Adversary

$\sigma_b^* = (b, r_b^*)$

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b^* = (b, r_b^*)$

σ_b^*

$\downarrow$

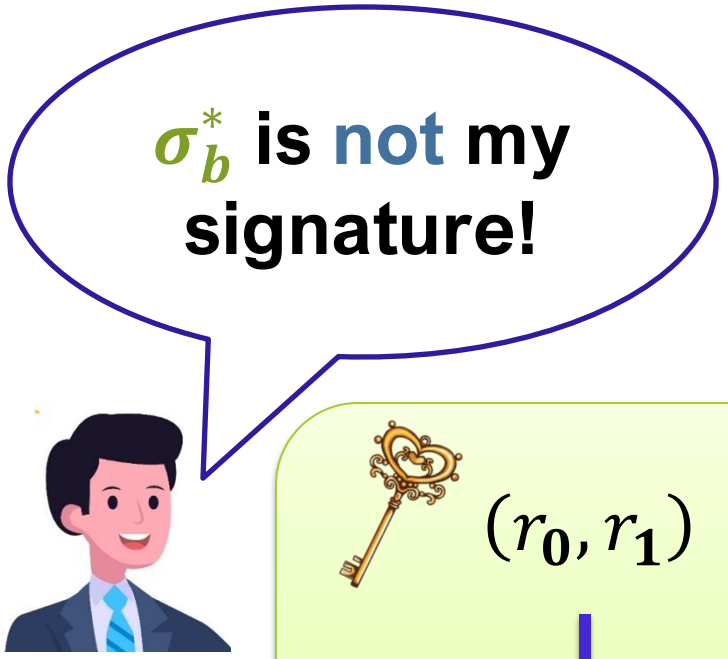
$h(r_b^*) = vk_b$

$\downarrow$

Accept

Recipient

Lamport Fail-Stop Signature



Adversary

$\sigma_b^* = (b, r_b^*)$

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b^* = (b, r_b^*)$

σ_b^*

$\downarrow$

$h(r_b^*) = vk_b$

$\downarrow$

Accept

Recipient

Lamport Fail-Stop Signature

Signer

Here is the proof:

Adversary

$\sigma_b^* = (b, r_b^*)$

(r_0, r_1)

$(h(r_0), h(r_1))$

(vk_0, vk_1)

$\sigma_b^* = (b, r_b^*)$

$\pi = (r_b, r_b^*)$

Recipient

Lamport Fail-Stop Signature

Here is the proof:

Adversary

$\sigma_b^* = (b, r_b^*)$

Signer

(r_0, r_1)

$(h(r_0), h(r_1))$

(vk_0, vk_1)

$\sigma_b^* = (b, r_b^*)$

$\pi = (r_b, r_b^*)$

Check

$h(r_b) = h(r_b^*)$

$r_b \neq r_b^*$

Accept

Recipient

Lamport Fail-Stop Signature

- The proof-of-forgery is a **collision** on h

Signer

(r_0, r_1)

$\downarrow$

$(h(r_0), h(r_1))$

$\downarrow \quad \downarrow$

(vk_0, vk_1)

$\sigma_b^* = (b, r_b^*)$

$\pi = (r_b, r_b^*)$

Check

$h(r_b) = h(r_b^*)$

$r_b \neq r_b^*$

$\downarrow$

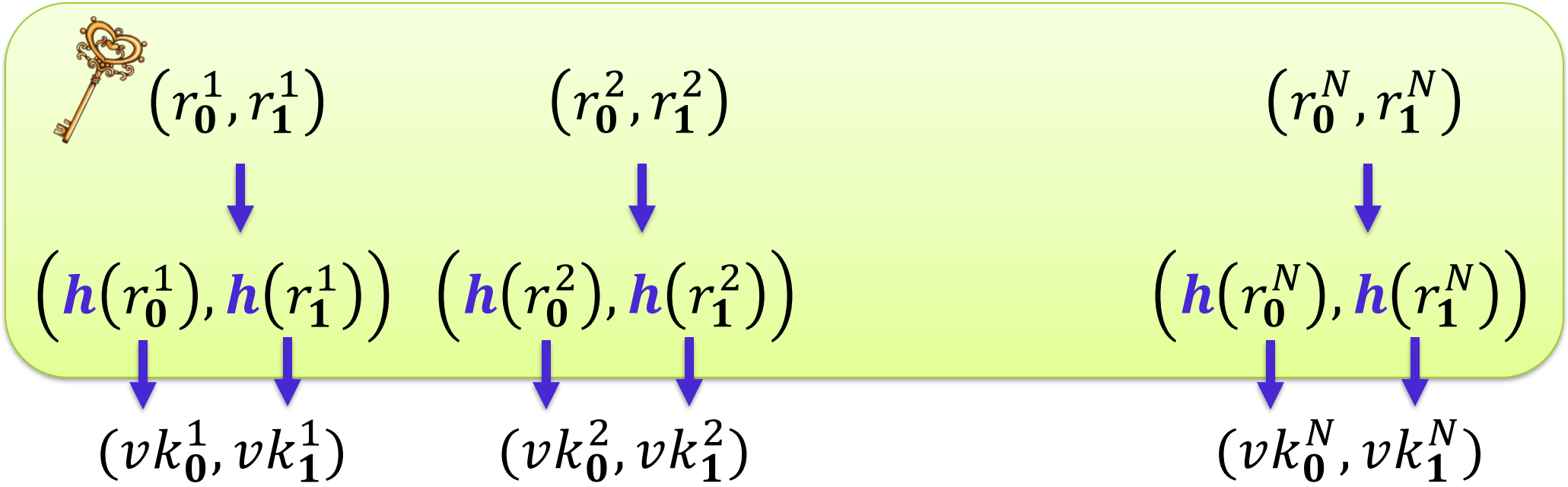
Accept

Recipient

Extending Lamport to Multiple-Use

- We want to sign on N bits
- Secret keys: (r_0^1, r_1^1),..., (r_0^N, r_1^N)

Signer



RSAC 2025 Conference

Extending Lamport to Multiple-Use

- The signer computes the pairs of secret keys (r_0^i, r_1^i) from the $\boldsymbol{Seed}$
- Secret keys: $\boldsymbol{Seed}$

Signer

$(vk_0^1, vk_1^1), \ldots, (vk_0^N, vk_1^N)$

Security Against Unbounded Adversaries

- We have only **one** **secret key** – the **seed**
- An **unbounded** **adversary** can **find** it!!

Signer

$(vk_0^1, vk_1^1), \ldots, (vk_0^N, vk_1^N)$

Challenges: Efficiency

- Fail-stop signatures assume **unbounded** adversaries, but this **prevents** us from constructing a **practical** fail-stop signature for **SPHINCS+**
- We introduce a new assumption inspired by **real-world risk** management practices against **adversaries**

“Fine-Grained” Adversarial Model

- **NIST** assumes that there are **different levels** of **adversaries**
- We **pay** in **efficiency** to **protect** against more **powerful adversaries**

	bit security	**signature (bytes)**
SPHINCS+-128s	128	7856
SPHINCS+-192s	192	16,224
SPHINCS+-256s	256	29,792

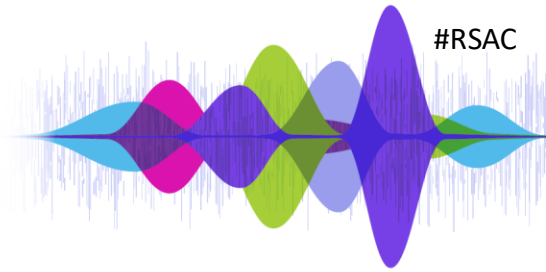
Our Results: Fail-Stop SPHINCS+

FSS.SPHINCS+ 128 bit with **256** bits of “fail-stop” security

is **smaller** and **faster** than **SPHINCS+ 256** bit

	pk (bytes)	sk (bytes)	sig (bytes)	Bit Sec Recipient	Bit Sec Signer
SPHINCS$^+$-128s	32	64	7856	128	128
FSS.SPHINCS-128s,$c_s = 2$ (log)	32	96	11,790	128	256
FSS.SPHINCS-128s,$c_s = 2$ (c_s sigs)	32	96	23,580	128	256
SPHINCS$^+$-256s	64	128	29,792	256	256

Summary



- We introduce fail-stop signatures to a post-quantum world
- We introduce the **“fine-grained”** adversarial model
- Our model allows us to **upgrade** the **security** while keeping **efficiency**

Pay a little now

You never know when your scheme might be broken

RSAC 2025 Conference

Many Voices.
One Community.

Thank you!

For more information:

https://eprint.iacr.org/2023/1754